NASA Advanced Modeling & Simulation (AMS) Seminar Series

NASA Ames Research Center, March 31, 2022

National Aeronautics and Space Administration

5 . . . 4 . . . 3 . . . 2 . . . 1 . . .

SPACE LAUNCH SYSTEM

March 31, 2022

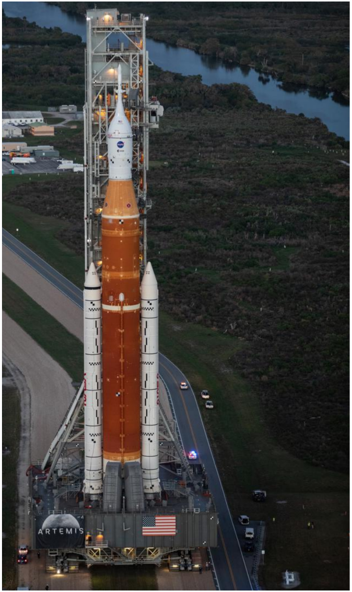
Advances in Launch Vehicle Ascent and Booster Separation CFD

Derek Dalle, Jamie Meeroff, Stuart Rogers, Aaron Burkhead, Guy Schauerhamer, Josh Diaz

Computational Aerosciences Branch
NASA Ames Research Center

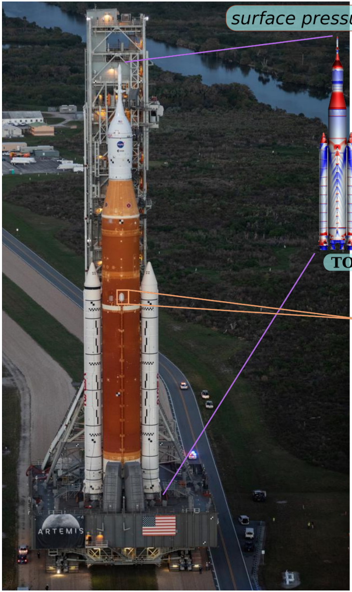
Ascent CFD

Ascent CFD

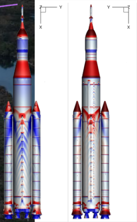


Ascent CFD

Ascent CFD



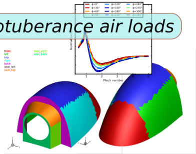
surface pressure databases



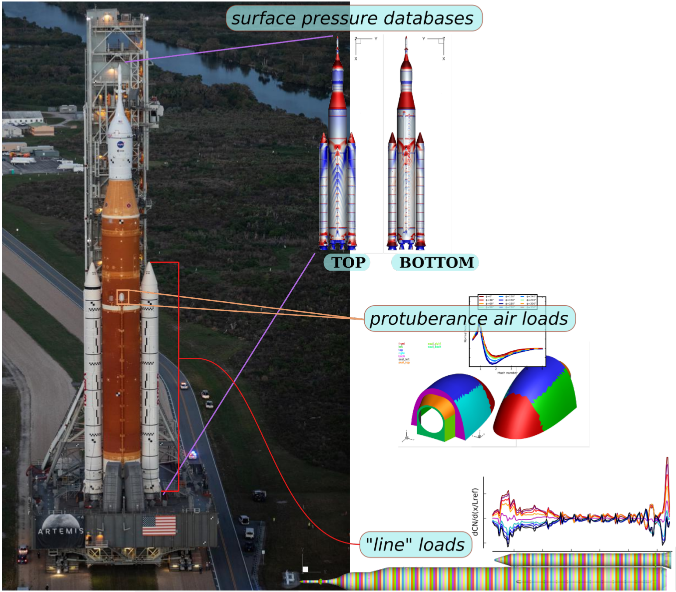
TOP

BOTTOM

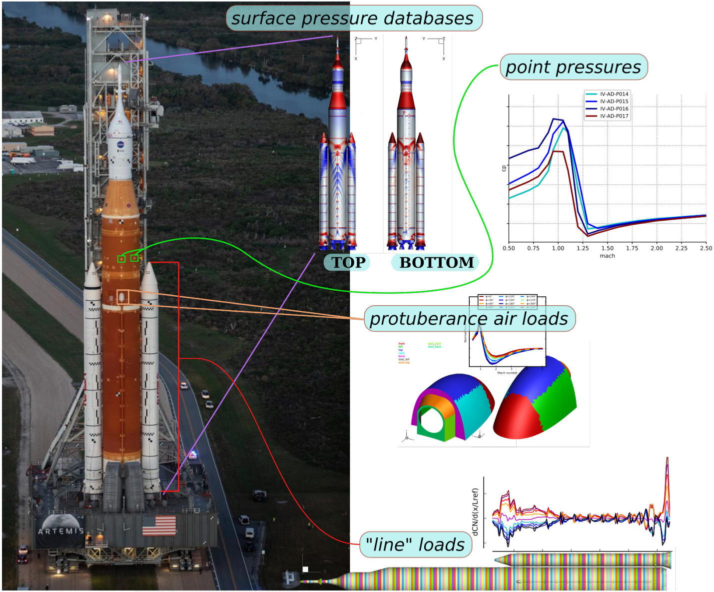
protuberance air loads



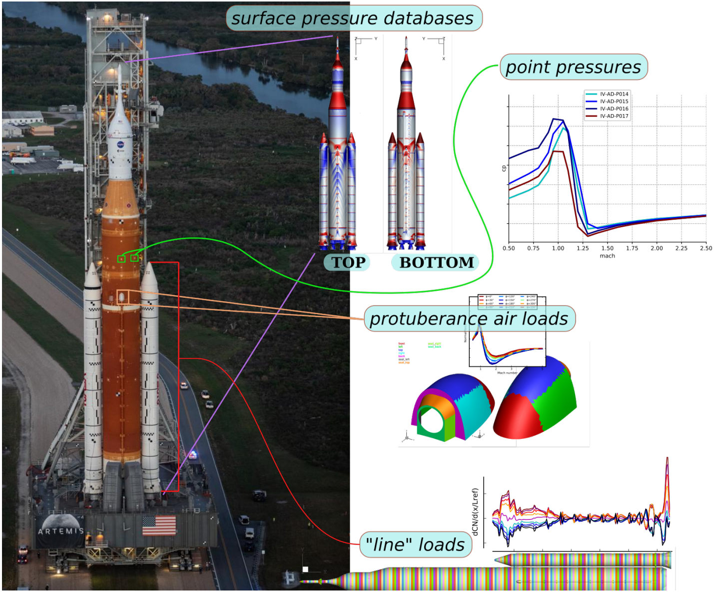
Ascent CFD


surface pressure databases
TOP
BOTTOM
protuberance air loads
"line" loads
dCN/d(x/Lref)
ARTEMIS

Ascent CFD

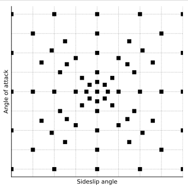

surface pressure databases
TOP
BOTTOM
point pressures
IV-AD-P014
IV-AD-P015
IV-AD-P016
IV-AD-P017
cp
0.50
0.75
1.00
1.25
1.50
1.75
2.00
2.25
2.50
mach
protuberance air loads
Mach number
"line" loads
dCN/d(x/Lref)
ARTEMIS

Ascent CFD



Run matrix

Mach 0.5

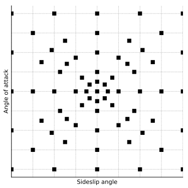


Mach 0.7

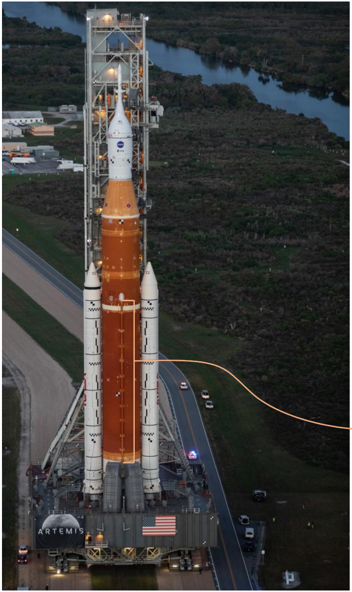
Mach 0.8

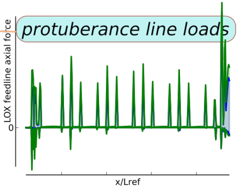
⋮

Mach 5.0



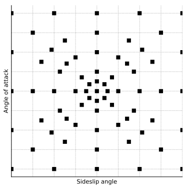
Ascent CFD (continued)





Run matrix

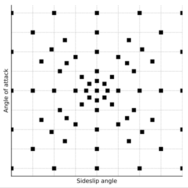
Mach 0.5



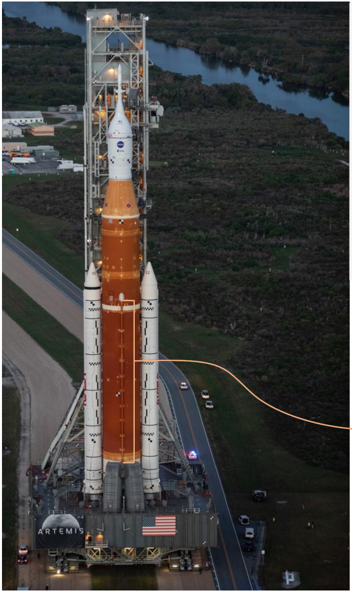
Mach 0.7
Mach 0.8

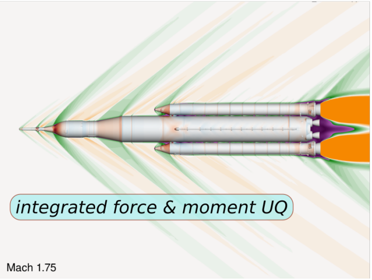
⋮

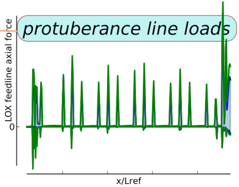
Mach 5.0



Ascent CFD (continued)

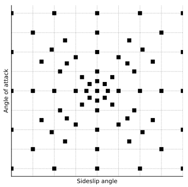






Run matrix

Mach 0.5

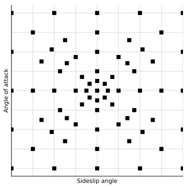


Mach 0.7

Mach 0.8

⋮

Mach 5.0

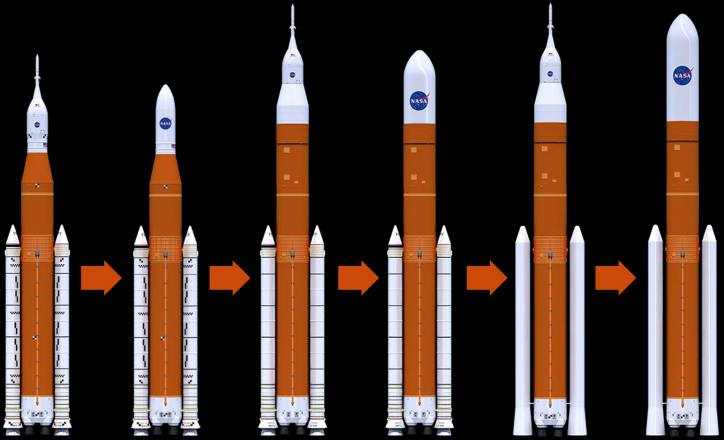


Do this for 6 (or more) configurations:

	SLS Block 1 Crew	SLS Block 1 Cargo	SLS Block 1B Crew	SLS Block 1B Cargo	SLS Block 2 Crew	SLS Block 2 Cargo
Payload to TLI/Moon	> 26 t (57k lbs)	> 26 t (57k lbs)	34–37 t (74k–81k lbs)	37–40 t (81k–88k lbs)	> 45 t (99k lbs)	> 45 t (99k lbs)
Payload Volume	N/A**	9,030 ft³ (256m³)	10,100 ft³ (286m³)**	18,970 ft³ (537 m³)	10,100 ft³ (286m³)**	34,910 ft³ (988 m³)
Maximum Thrust	8.8M lbs	8.8M lbs	8.8M lbs	8.8M lbs	11.9M lbs	11.9M lbs

Trans-Lunar Injection (TLI) is a propulsive maneuver used to set a spacecraft on a trajectory that will cause it to arrive at the Moon. A spacecraft performs **TLI** to begin a lunar transfer from a low circular parking orbit around Earth.

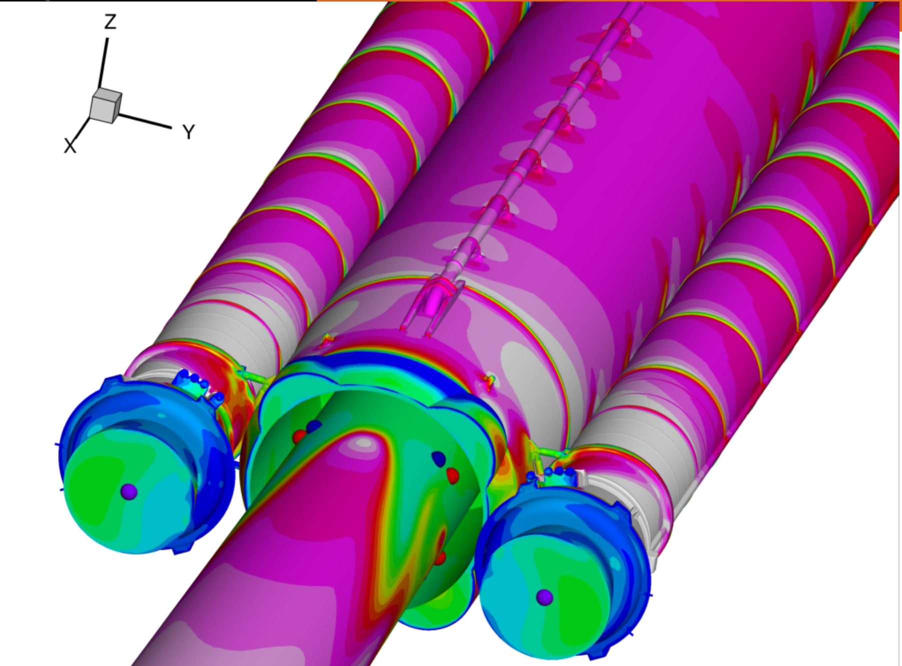
The numbers depicted here indicate the mass capability at the Trans-Lunar Injection point.



SLS Block 1 Crew | SLS Block 1 Cargo | SLS Block 1B Crew | SLS Block 1B Cargo | SLS Block 2 Crew | SLS Block 2 Cargo

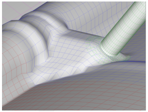
** Not including Orion/Service Module volume

Run CFD on wind tunnel models, too

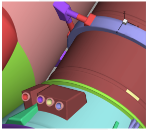


History, from Space Shuttle to 2022

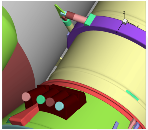
Space Shuttle, c. 2006



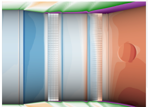
Artemis I DAC3, c. 2014

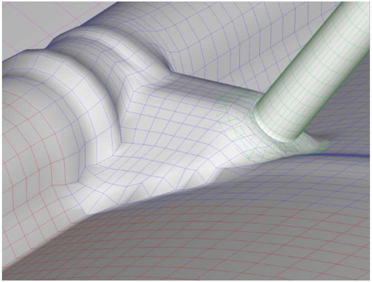


Artemis I VAC1, c. 2017

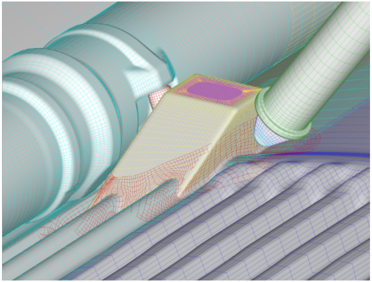


Artemis I FRAC1, c. 2022

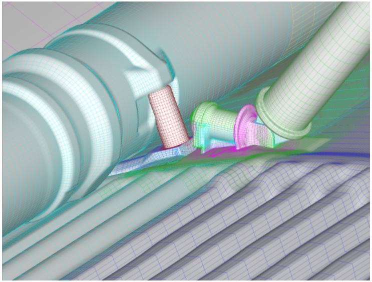




1990s



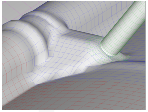
c. 2003



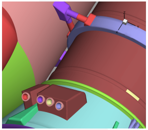
c. 2006

History, from Space Shuttle to 2022

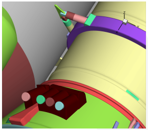
Space Shuttle, c. 2006



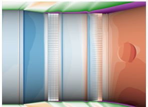
Artemis I DAC3, c. 2014

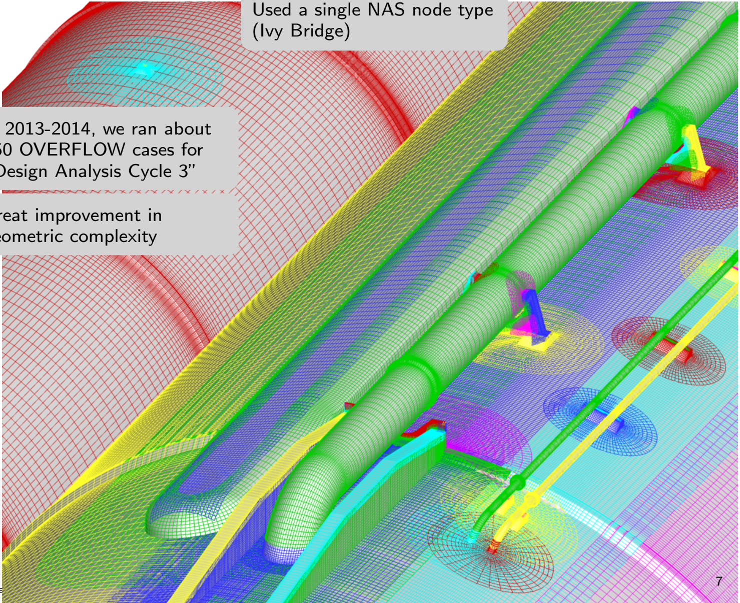


Artemis I VAC1, c. 2017



Artemis I FRAC1, c. 2022





Used a single NAS node type (Ivy Bridge)

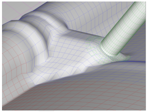
In 2013-2014, we ran about 850 OVERFLOW cases for "Design Analysis Cycle 3"

Great improvement in geometric complexity

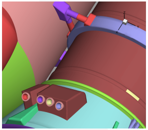
www.na

History, from Space Shuttle to 2022

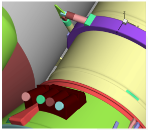
Space Shuttle, c. 2006



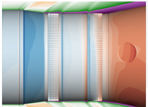
Artemis I DAC3, c. 2014

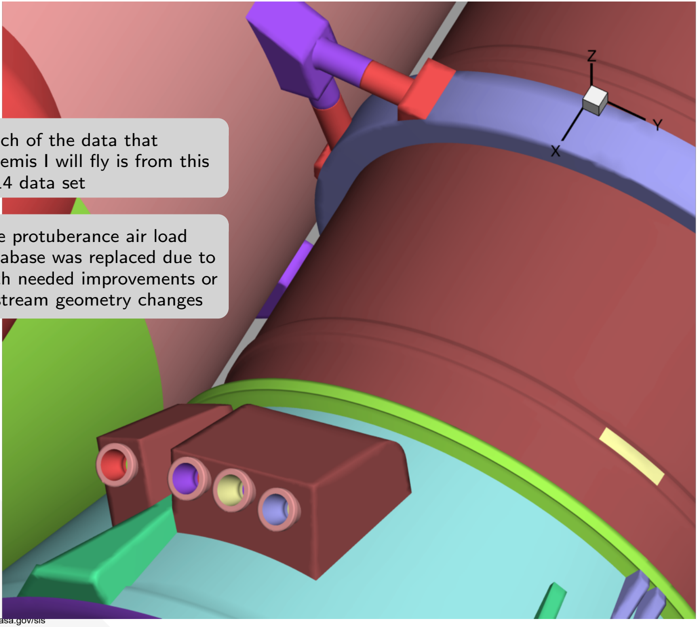


Artemis I VAC1, c. 2017



Artemis I FRAC1, c. 2022



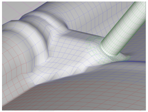


Much of the data that Artemis I will fly is from this 2014 data set

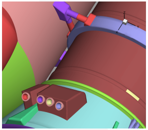
The protuberance air load database was replaced due to both needed improvements or upstream geometry changes

History, from Space Shuttle to 2022

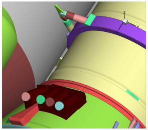
Space Shuttle, c. 2006



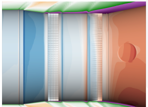
Artemis I DAC3, c. 2014

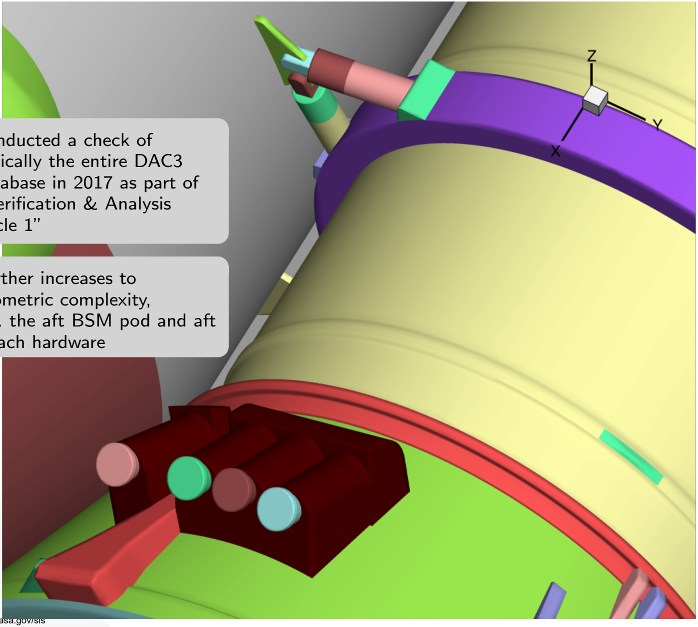


Artemis I VAC1, c. 2017



Artemis I FRAC1, c. 2022



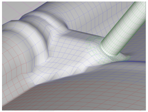


Conducted a check of basically the entire DAC3 database in 2017 as part of "Verification & Analysis Cycle 1"

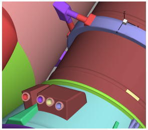
Further increases to geometric complexity, e.g. the aft BSM pod and aft attach hardware

History, from Space Shuttle to 2022

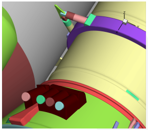
Space Shuttle, c. 2006



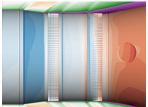
Artemis I DAC3, c. 2014

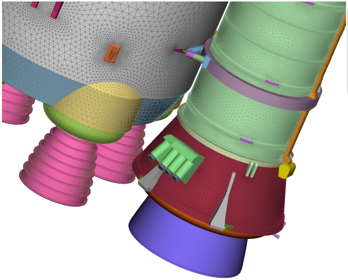


Artemis I VAC1, c. 2017

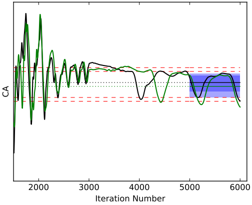


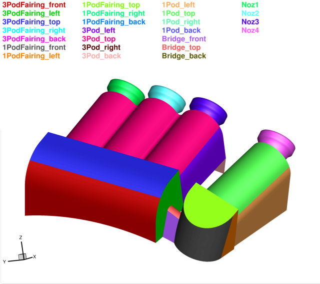
Artemis I FRAC1, c. 2022



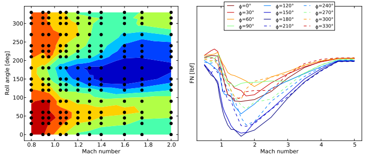


Created the CAPE software package to improve flexibility, speed, and reproducibility
We were able to run 1300 cases in both FUN3D and OVERFLOW





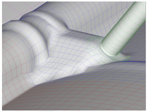
Instrument and check iterative convergence while running



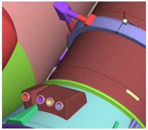
Attempt to make meaningful plots

History, from Space Shuttle to 2022

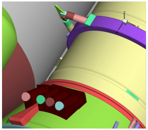
Space Shuttle, c. 2006



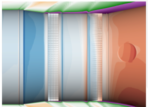
Artemis I DAC3, c. 2014



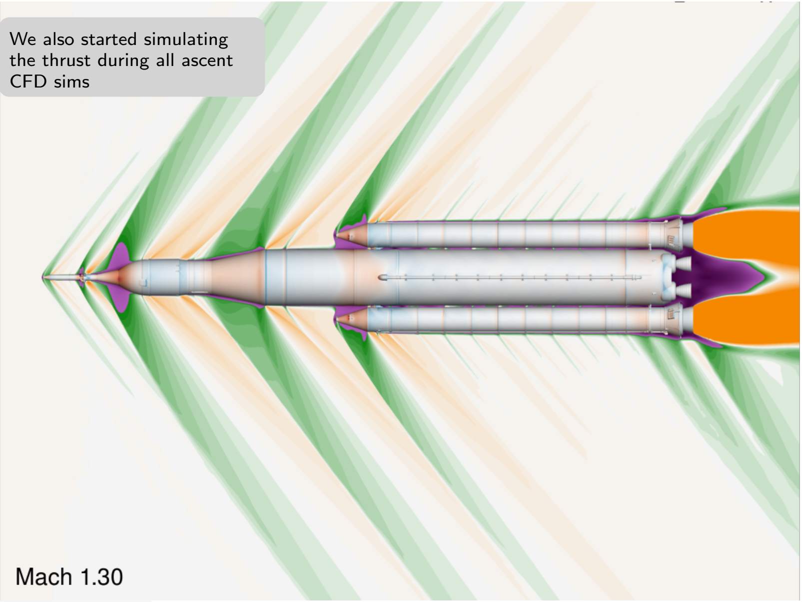
Artemis I VAC1, c. 2017



Artemis I FRAC1, c. 2022

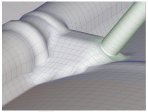


We also started simulating the thrust during all ascent CFD sims

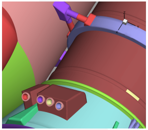


History, from Space Shuttle to 2022

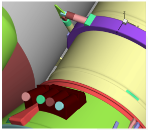
Space Shuttle, c. 2006



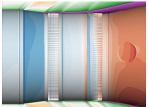
Artemis I DAC3, c. 2014

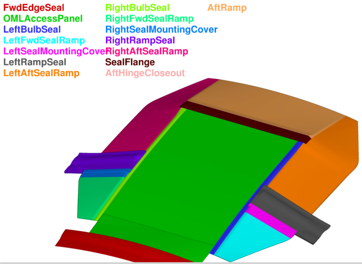


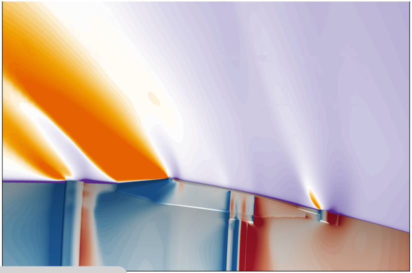
Artemis I VAC1, c. 2017



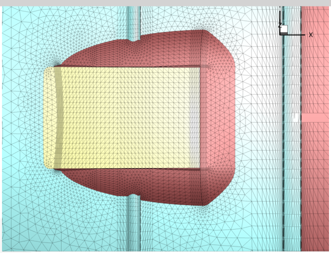
Artemis I FRAC1, c. 2022



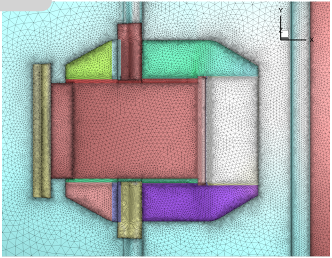




Lots of late-breaking questions. This example is the umbilical that connects the crew module to the service module, which changed a great deal



SLS-10008 CSM umbilical mesh



2020 geometry

Tool #1: treat CFD like software dev

Each of these boxes is a git repository used in our analysis of one Artemis I task

`grid/SLS-10F-009`
- original CAD files
- cleaned up CAD files
- grid generation files
- prepared grid files
- some CFD config files

`run/10009/f3_frac0`
- prepared grid files
- template CFD inputs
- CAPE files
- actual CFD runs

`run/10009/of_frac0`
- prepared grid files
- template CFD inputs
- CAPE files
- actual CFD runs

`run/10009/f3_frac0/data/poweron01`
- data extracted by pyfun

`run/10009/f3_frac0/data/poweron02`
- data extracted by pyfun

`run/10009/f3_frac0/data/clean01`
- data extracted by pyfun

`run/10009/of_frac0/data/clean01`
- data extracted by pyfun

`data/SLS-10-D-AFA`
- data from pyfun
- python files
- fixed-up databases

`data/SLS-10-D-ALA`
- data from pyfun
- python files
- fixed-up databases

Key aspect: use code and/or command line to generate inputs for each CFD case; no clicking to copy files manually!

Tool #1: treat CFD like software dev

Each of these boxes is a git repository used in our analysis of one Artemis I task

grid/SLS-10F-009
- original CAD files
- cleaned up CAD files
- grid generation files
- prepared grid files
- some CFD config files

run/10009/f3_frac0
- prepared grid files
- template CFD inputs
- CAPE files
- actual CFD runs

run/10009/of_frac0
- prepared grid files
- template CFD inputs
- CAPE files
- actual CFD runs

run/10009/f3_frac0/data/poweron01
- data extracted by pyfun

run/10009/f3_frac0/data/poweron02
- data extracted by pyfun

run/10009/f3_frac0/data/clean01
- data extracted by pyfun

run/10009/of_frac0/data/clean01
- data extracted by pyfun

data/SLS-10-D-AFA
- data from pyfun
- python files
- fixed-up databases

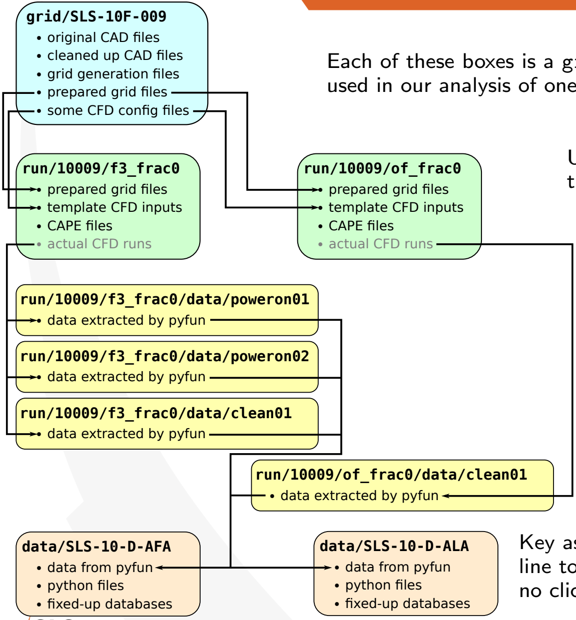
data/SLS-10-D-ALA
- data from pyfun
- python files
- fixed-up databases

Using version control software like this has certain implications:

- No spreadsheets, rich-text word docs, etc.
- Everything that changes w/ time should be text
- Can be a challenge for new team members to learn

Key aspect: use code and/or command line to generate inputs for each CFD case; no clicking to copy files manually!

Tool #1: treat CFD like software dev



Each of these boxes is a git repository used in our analysis of one Artemis I task

Using version control software like this has certain implications:

- No spreadsheets, rich-text word docs, etc.
- Everything that changes w/ time should be text
- Can be a challenge for new team members to learn
- Avoid having one HUGE repo where all the work is done
- This requires some planning

Key aspect: use code and/or command line to generate inputs for each CFD case; no clicking to copy files manually!

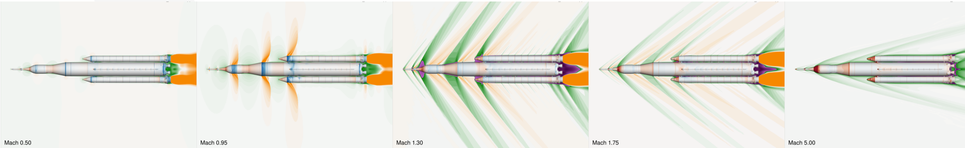
Tool #2: CAPE

Use a single tool to set up, run, backup, and extract data from each case

Running

Example: set up and submit 10 OVERFLOW jobs at Mach 1.75

```
$ pyover --re m1.75 -n 10
```



Tool #2: CAPE

Use a single tool to set up, run, backup, and extract data from each case

Running

Example: set up and submit 10 OVERFLOW jobs at Mach 1.75

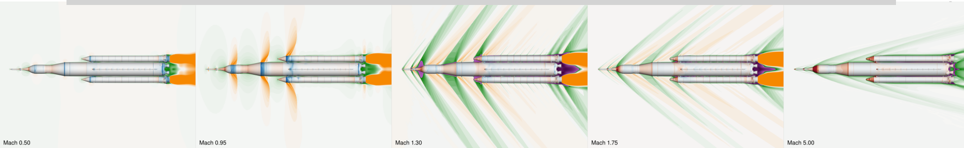
```
$ pyover --re m1.75 -n 10
```

Checking status

Example: Check status of FUN3D jobs at Mach 1.75

```
$ pyfun --re m1.75 -c
Case Config/Run Directory     Status  Iterations  Que CPU Time
---- ----------------------- ------- ----------- --- --------
81   poweron/m1.75a0.0r000.0 RUNNING 4237/5000   R    11273.7
82   poweron/m1.75a4.0r000.0 QUEUE   3000/4000   Q     2633.1
83   poweron/m1.75a4.0r090.0 PASS    5000/5000   .    10743.3

PASS=1, RUNNING=1, QUEUE=1,
```



Tool #2: CAPE

Use a single tool to set up, run, backup, and extract data from each case

Running

Example: set up and submit 10 OVERFLOW jobs at Mach 1.75

```
$ pyover --re m1.75 -n 10
```

Generate report

Example: generate LaTeX report for cases 79 and 402

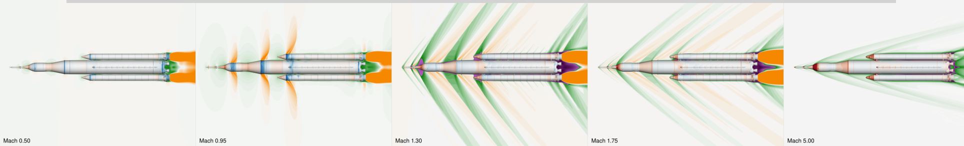
```
$ pycart -I 79,402 --report
```

Checking status

Example: Check status of FUN3D jobs at Mach 1.75

```
$ pyfun --re m1.75 -c
Case Config/Run Directory     Status  Iterations  Que CPU Time
---- ----------------------- ------- ----------- --- --------
81   poweron/m1.75a0.0r000.0 RUNNING 4237/5000   R    11273.7
82   poweron/m1.75a4.0r000.0 QUEUE   3000/4000   Q     2633.1
83   poweron/m1.75a4.0r090.0 PASS    5000/5000   .    10743.3

PASS=1, RUNNING=1, QUEUE=1,
```



Tool #2: CAPE

Use a single tool to set up, run, backup, and extract data from each case

Running

Example: set up and submit 10 OVERFLOW jobs at Mach 1.75

```
$ pyover --re m1.75 -n 10
```

Generate report

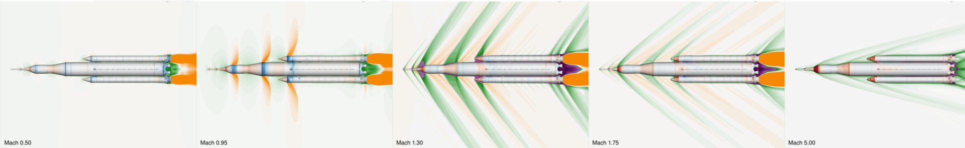
Example: generate LaTeX report for cases 79 and 402

```
$ pycart -I 79,402 --report
```

Collect forces and moments

Example: update F&M database for negative-sideslip cases

```
$ pyfun --cons "beta<0" --fm
```



Tool #2: CAPE

Use a single tool to set up, run, backup, and extract data from each case

Running

Example: set up and submit 10 OVERFLOW jobs at Mach 1.75

```
$ pyover --re m1.75 -n 10
```

Generate report

Example: generate LaTeX report for cases 79 and 402

```
$ pycart -I 79,402 --report
```

Collect forces and moments

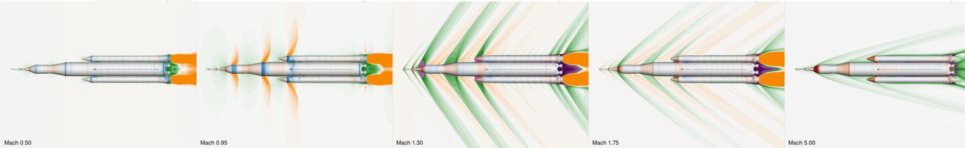
Example: update F&M database for negative-sideslip cases

```
$ pyfun --cons "beta<0" --fm
```

Extract protuberance air loads

Example: get patch loads for components starting with “M”

```
$ pyover --triqfm "M*"
```



Tool #2: CAPE

Use a single tool to set up, run, backup, and extract data from each case

Running

Example: set up and submit 10 OVERFLOW jobs at Mach 1.75

```
$ pyover --re m1.75 -n 10
```

Generate report

Example: generate LaTeX report for cases 79 and 402

```
$ pycart -I 79,402 --report
```

Collect forces and moments

Example: update F&M database for negative-sideslip cases

```
$ pyfun --cons "beta<0" --fm
```

Extract protuberance air loads

Example: get patch loads for components starting with "M"

```
$ pyover --triqfm "M*"
```

Archiving

Example: create backup and delete large files from working copy

```
$ pyover --archive
```

... and recover

Example: retrieve case 309 from archive for more analysis

```
$ pyover --unarchive -I 309
```

Tool #3: robust CFD solvers

FUN3D

Fully Unstructured Navier-Stokes

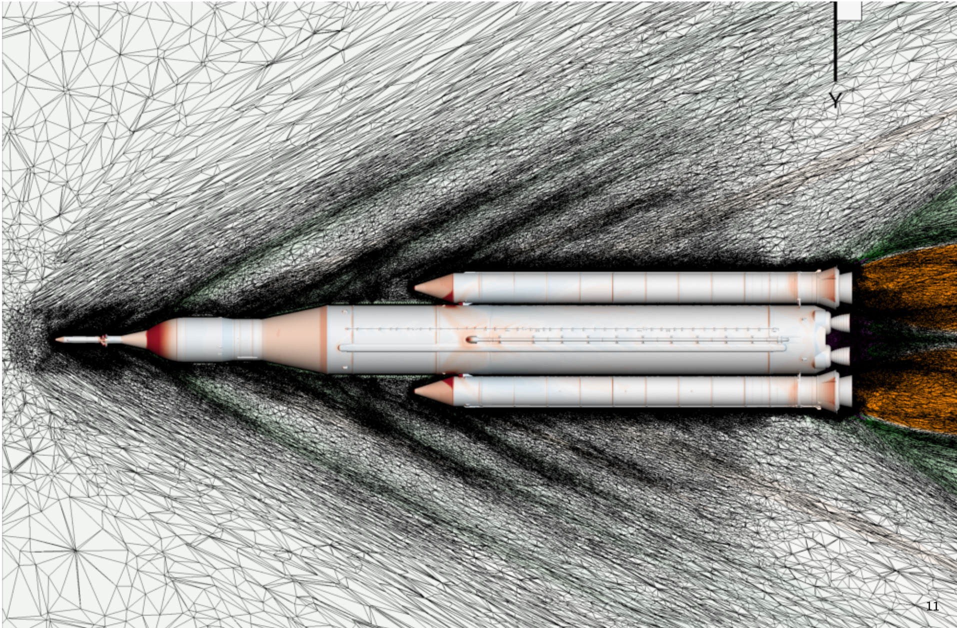
https://fun3d.larc.nasa.gov

OVERFLOW 2

https://overflow.larc.nasa.gov/

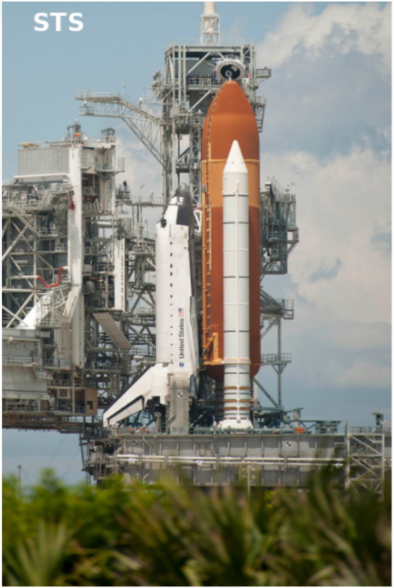
https://www.nas.nasa.gov/publications/software/docs/cart3d/

Tool #4: mesh adaptation

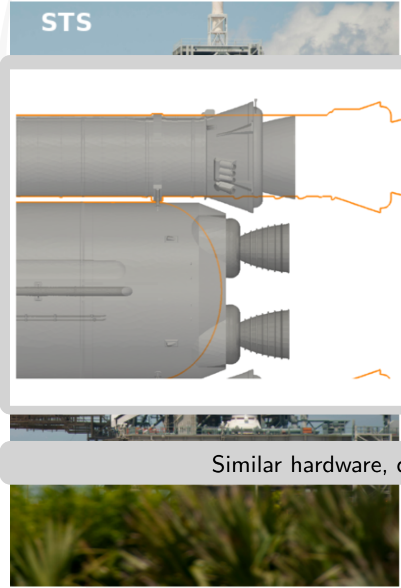


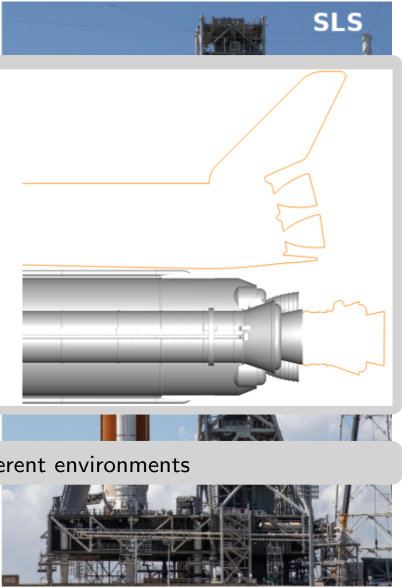
Booster Separation

Motivation



Motivation

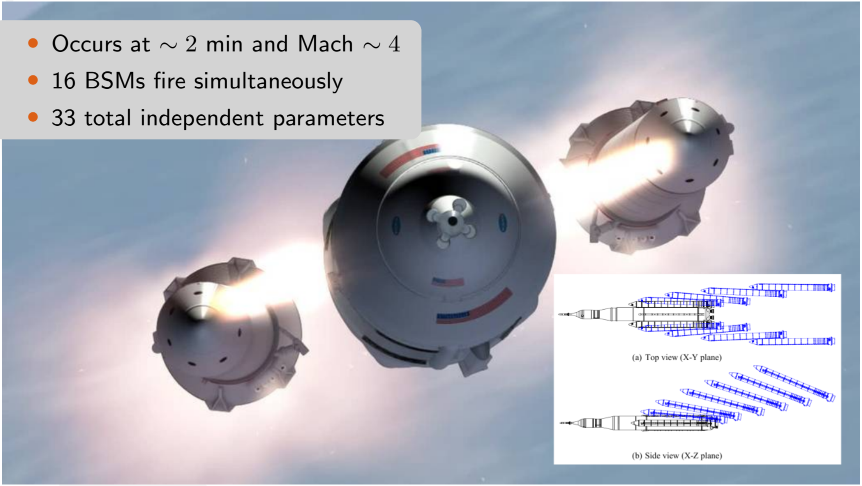


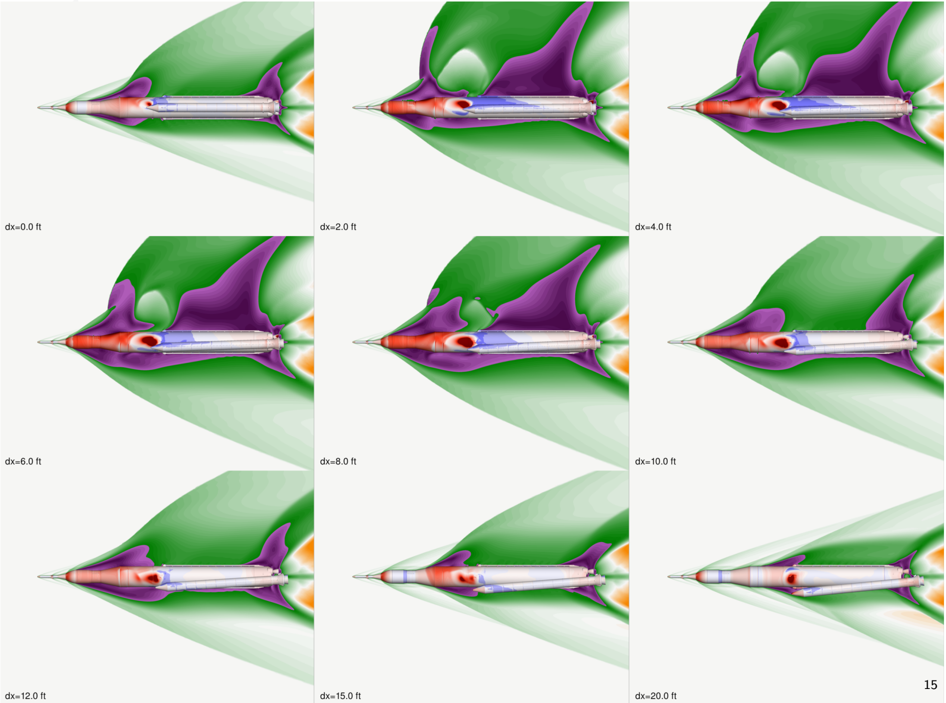


Similar hardware, different environments

Booster Separation Overview

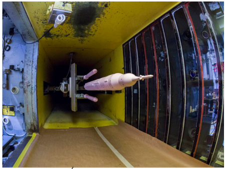
- Occurs at ~ 2 min and Mach ~ 4
- 16 BSMs fire simultaneously
- 33 total independent parameters

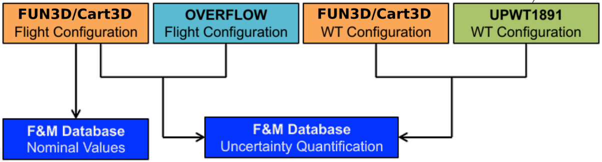



dx=0.0 ft
dx=2.0 ft
dx=4.0 ft
dx=6.0 ft
dx=8.0 ft
dx=10.0 ft
dx=12.0 ft
dx=15.0 ft
dx=20.0 ft

Booster Separation Database

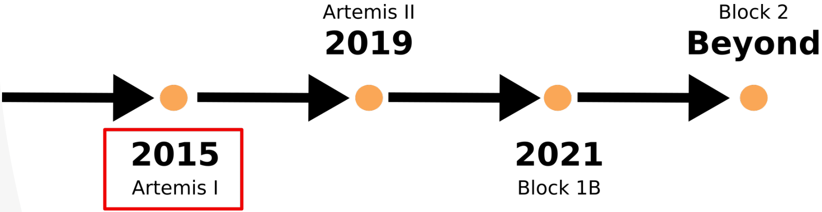
- Shuttle used wind tunnel to build database
- SLS uses CFD for nominals (cost, physics)
- Wind tunnel still used for uncertainty





Forces and moments serve as inputs for GN&C simulations

Booster Separation Timeline



- Cart3D
- Limited to 8 dimensions
- Initial development of CAPE software
- Over 20k cases run

Simplifying the Problem

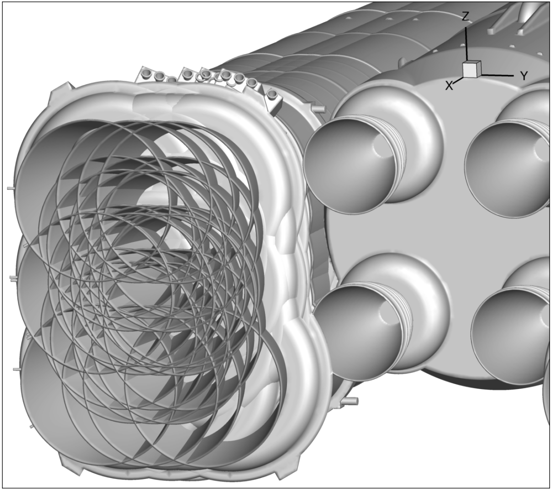
8-dimensional run matrix:

Variable	Description
dx	SRB axial translation
dy	SRB outward translation
dz	SRB vertical translation
dpsi	SRB yaw (rel. to core)
dtheta	SRB pitch
alpha	CORE angle of attack
beta	CORE sideslip angle
CTBSM	BSM thrust coefficient

Other variables held constant:

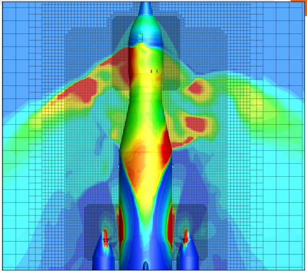
Variable	Description
dphi	SRB roll
mach	CORE Mach number
CTCSE	CORE engine thrust
CTSRB	SRB thrust

Full run matrix: $\mathcal{O}$(10k) cases

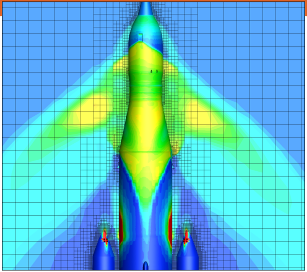


All SRB positions simulated at dx=6 ft in Artemis I database

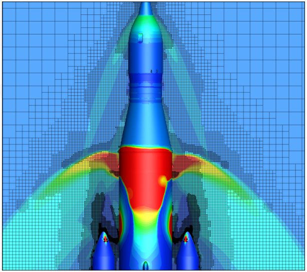
Artemis I B. Separation with Cart3D



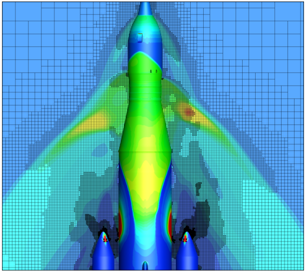
Manually specified mesh



Mesh generated automatically with `autoInputs`

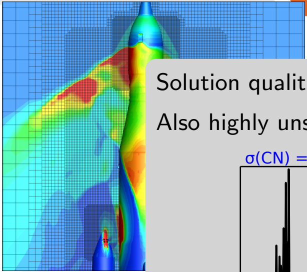


Adapted mesh using 10 cycles from coarse initial mesh

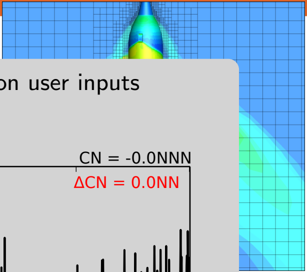


Adapted mesh using 6 cycles from finer initial mesh

Artemis I B. Separation with Cart3D



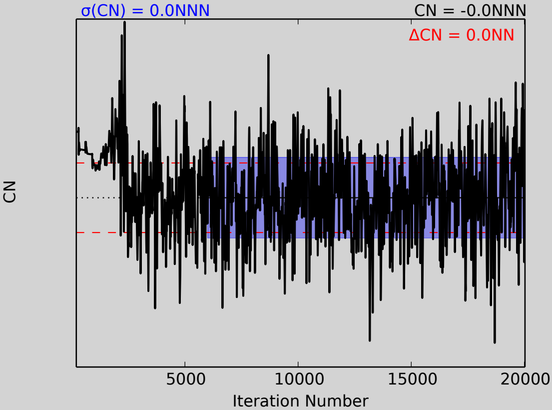
Manually sp

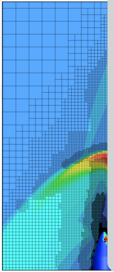


lly with

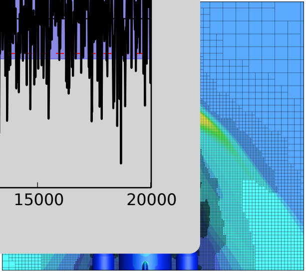
Solution qualitatively dependent on user inputs

Also highly unsteady:





Adapted mesh using 10 cycles from coarse initial mesh



Adapted mesh using 6 cycles from finer initial mesh

Booster Separation Timeline

Artemis II
2019

Block 2
Beyond

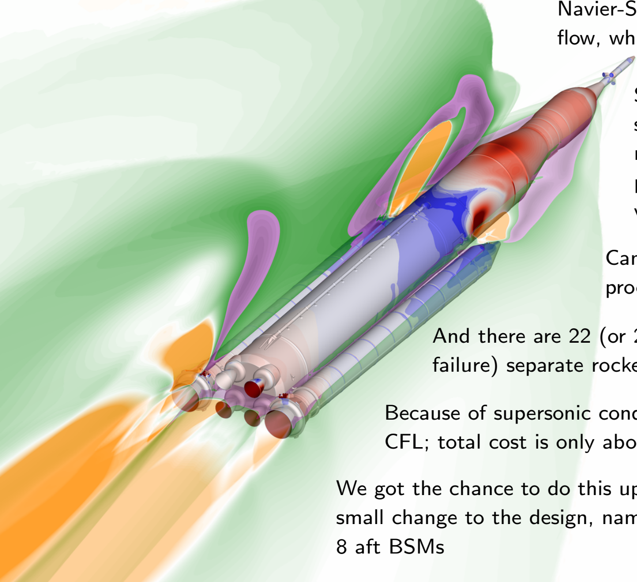
2015
Artemis I

2021
Block 1B

- FUN3D
- *Smarter* run matrix
- Expanded variable set (13 dimensions)
- Over 14k CFD cases run

Artemis II B. Separation with FUN3D

Example from relatively normal
dx=4 ft, CSE1 failure



Figured out how to use a full Navier-Stokes solver (FUN3D) for this flow, which required some luck

Setup is very complicated; three separate bodies, which are rotated and translated into place. Then create case-specific volume grids

Can handle slight intersections, processed before volume grid

And there are 22 (or 21 when modeling an engine failure) separate rockets firing

Because of supersonic conditions, can use a huge ($\sim$200) CFL; total cost is only about 4x what we had from Cart3D

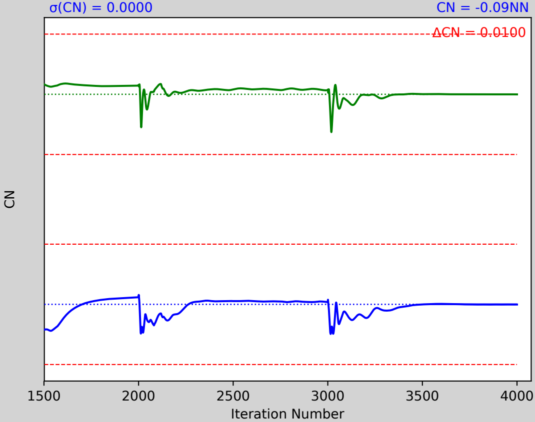
We got the chance to do this update because Artemis II has a small change to the design, namely a 15° rotation of 6 of the 8 aft BSMs

Artemis II B. Separation with FUN3D

Example from relatively normal
dx=4 ft, CSE1 failure

Solutions now very steady

Showing an example with normal force on both SRBs



Figured out how to use a full
…UN3D) for this
…me luck

…omplicated; three
…s, which are
…nslated into
…eate case-specific

…intersections,
…volume grid

…ling an engine

… a huge (∼200)
… had from Cart3D

…Artemis II has a
small change to the design, namely a 15° rotation of 6 of the
8 aft BSMs

FUN3D Procedure

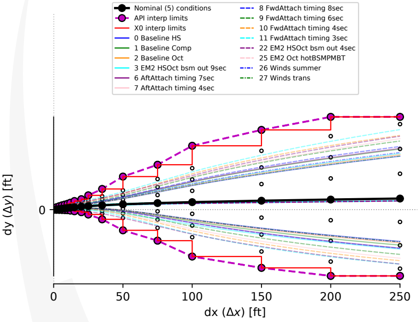
- Despite the complexity of the flow, this is a "computationally easy" problem
- Run FUN3D steady-state and aggressively; CFL ramps up to 200
- Nominally runs 4000 iterations for BSM-on, 3000 for BSM-off

Procedure

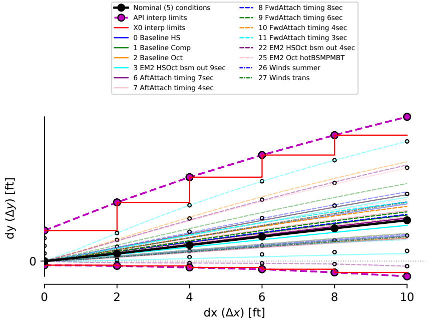
1. Move SRB surfaces into place
2. Run `intersect` and patch up if necessary
3. Generate volume mesh using `aflr3` (use one node)
4. Resubmit job with more than 1 compute node
5. Run first 1000 iterations, then adapt mesh to 19M points
6. Run next 2000 (1000 for BSM-off) iterations, then adapt mesh to 25M points
7. Run last 1000 iterations

- Initial mesh has about 12M points
- Using FUN3D-internal feature-based adaptation
- If mesh fails, we have sequence of AFLR3 settings to try
- If FUN3D fails, we have series of other CFL schedules to try

Artemis II Run Matrix



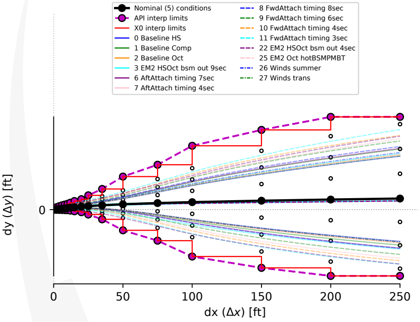
dy run matrix points and trajectory limits vs *dx*



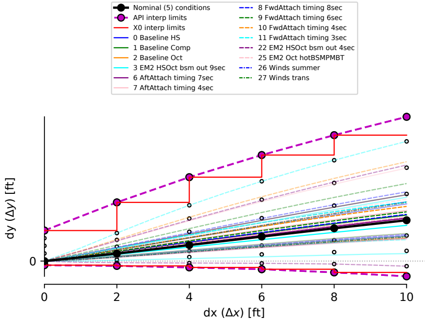
. . . focus on first 10 ft of *dx*

- For Artemis I, we had a manual run matrix of just min/mean/max conditions
- CFD for Artemis II was closer to the actual conditions expected in flight

Artemis II Run Matrix



dy run matrix points and trajectory limits vs *dx*



. . . focus on first 10 ft of *dx*

- For Artemis I, we had a manual run matrix of just min/mean/max conditions
- CFD for Artemis II was closer to the actual conditions expected in flight

- Despite all these improvements, UQ was not significantly better than Artemis I

Booster Separation Timeline

Artemis II
2019

Block 2
Beyond

2015
Artemis I

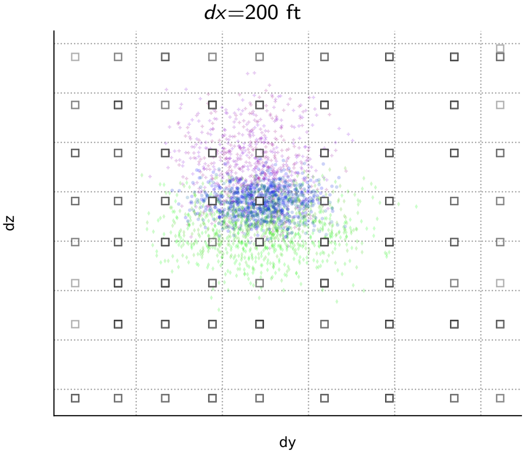
2021
Block 1B

- FUN3D
- Correlated run matrix (covariance)
- Reduced number of variables (lessons learned)
- Similar computational cost to Artemis II, fewer failures

Block 1B: Better Run Matrix

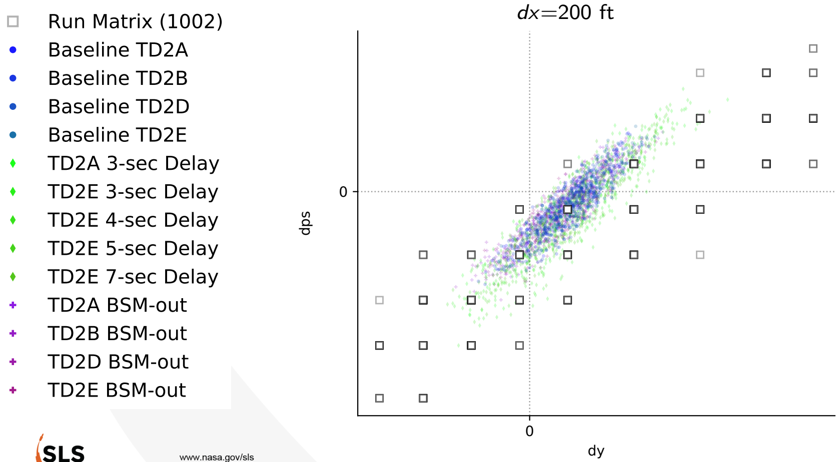
- Plots of SRB *dy* vs *dz*
- Run matrix tailored to 13 GN&C scenarios in three families
- *dy* and *dz* mostly uncorrelated
- Typical rectangular run matrix appropriate

Run Matrix (1002)
Baseline TD2A
Baseline TD2B
Baseline TD2D
Baseline TD2E
TD2A 3-sec Delay
TD2E 3-sec Delay
TD2E 4-sec Delay
TD2E 5-sec Delay
TD2E 7-sec Delay
TD2A BSM-out
TD2B BSM-out
TD2D BSM-out
TD2E BSM-out



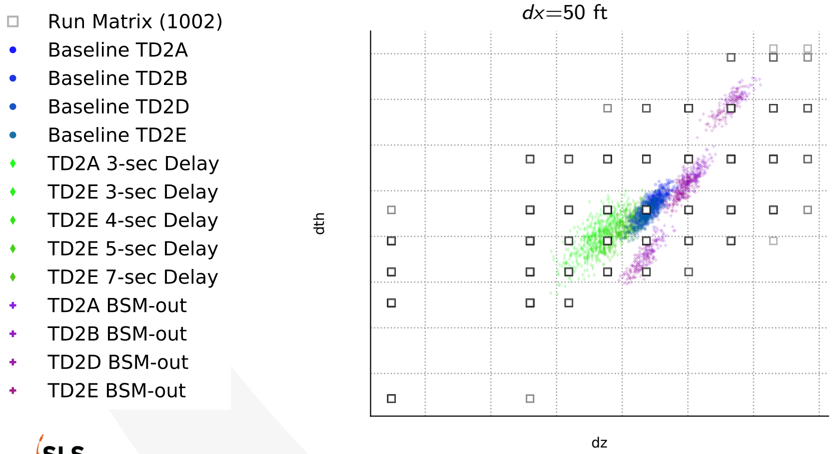
Key Run Matrix Improvements

- Plots of SRB *dy* vs *dpsi*
- Run matrix tailored to 13 GN&C scenarios in three families
- *dy* and *dpsi* highly correlated; higher yaw leads to higher side forces, which push the SRBs to higher *dy*

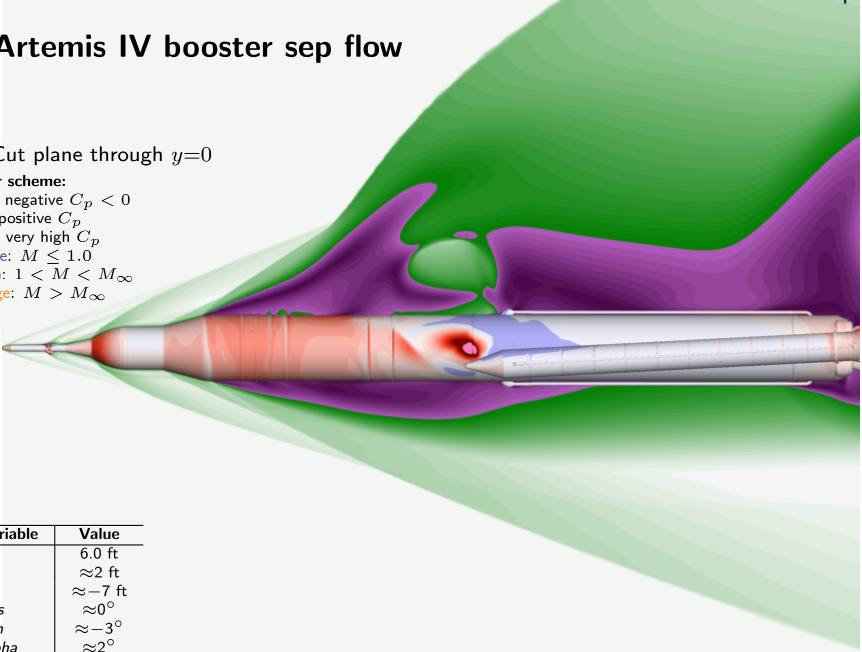


Key Run Matrix Improvements

- Plots of SRB *dz* vs *dtheta*
- Run matrix tailored to 13 GN&C scenarios in three families
- *dz* and *dtheta* highly correlated; more pitch down leads to negative normal forces, which push the SRBs to lower *dz*



Artemis IV booster sep flow

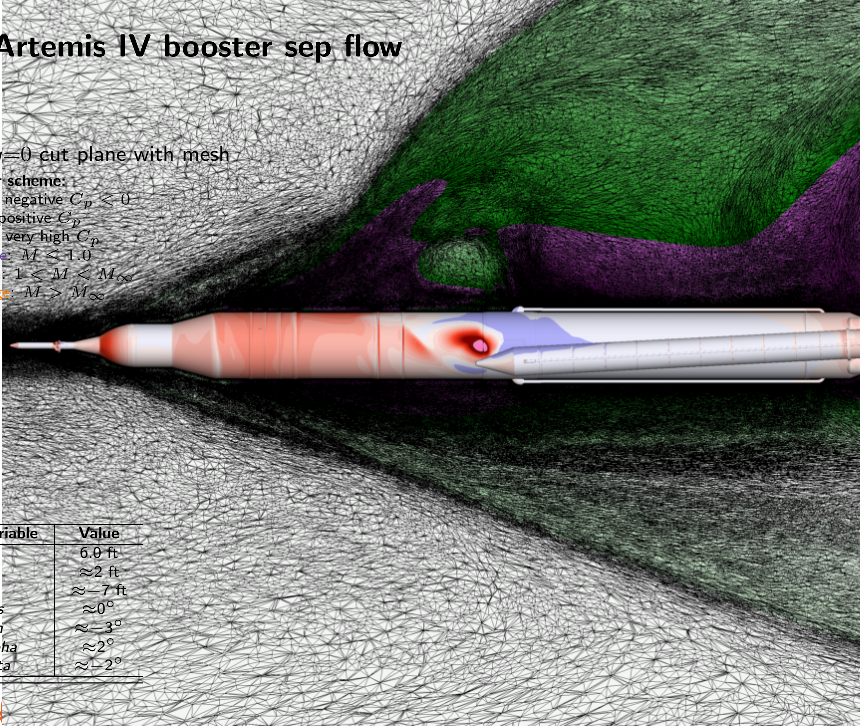


1. Cut plane through $y=0$

Color scheme:
blue: negative $C_p < 0$
red: positive C_p
pink: very high C_p
purple: $M \leq 1.0$
green: $1 < M < M_\infty$
orange: $M > M_\infty$

Variable	Value
dx	6.0 ft
dy	≈2 ft
dz	≈−7 ft
dps	≈0°
dth	≈−3°
alpha	≈2°
beta	≈−2°

Artemis IV booster sep flow



2. $y=0$ cut plane with mesh

Color scheme:

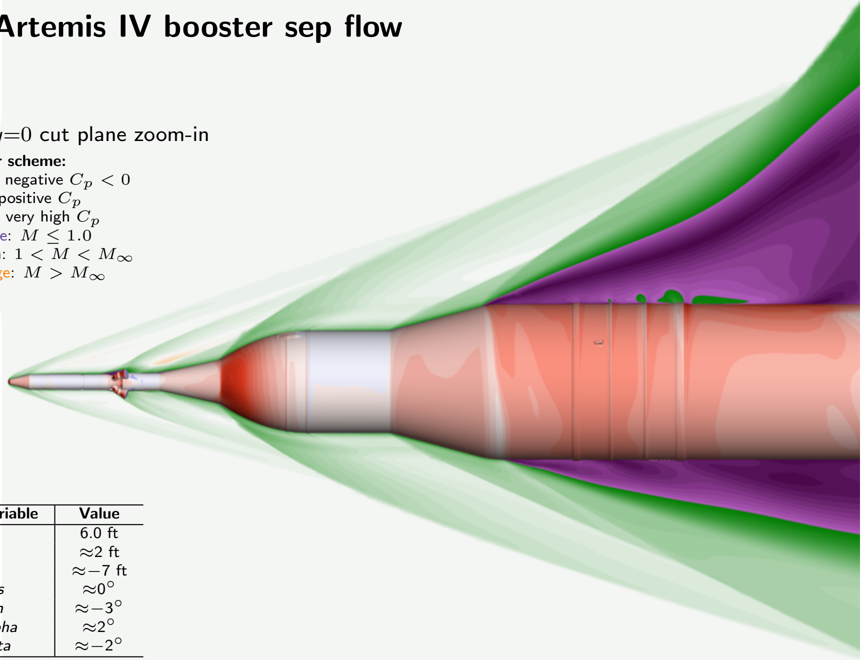
blue: negative $C_p < 0$
red: positive C_p
pink: very high C_p
purple: $M \leq 1.0$
green: $1 < M < M_\infty$
orange: $M > M_\infty$

Variable	Value
dx	6.0 ft
dy	$\approx$2 ft
dz	≈ -7 ft
dps	$\approx 0^\circ$
dth	$\approx -3^\circ$
alpha	$\approx 2^\circ$
beta	$\approx -2^\circ$

Artemis IV booster sep flow

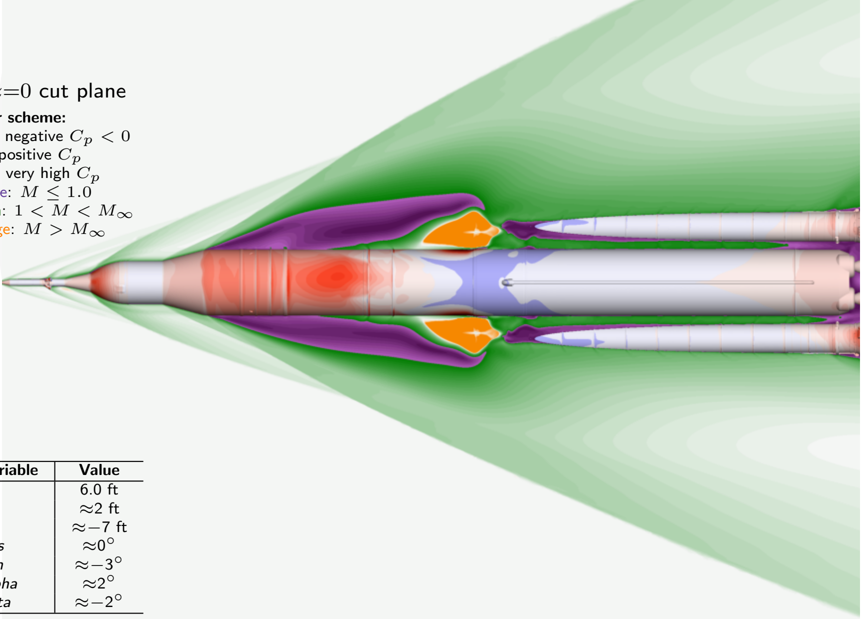
3. y=0 cut plane zoom-in

Color scheme:
blue: negative $C_p < 0$
red: positive C_p
pink: very high C_p
purple: $M \leq 1.0$
green: $1 < M < M_\infty$
orange: $M > M_\infty$



Variable	Value
dx	6.0 ft
dy	≈2 ft
dz	≈−7 ft
dps	≈0°
dth	≈−3°
alpha	≈2°
beta	≈−2°

Artemis IV booster sep flow

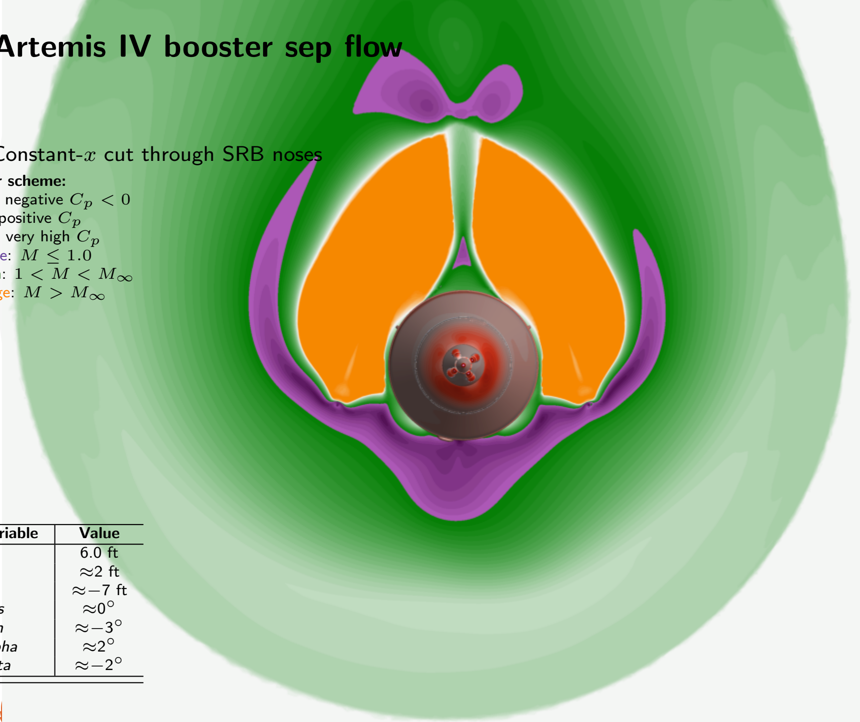


4. z=0 cut plane

Color scheme:
blue: negative $C_p < 0$
red: positive C_p
pink: very high C_p
purple: $M \leq 1.0$
green: $1 < M < M_\infty$
orange: $M > M_\infty$

Variable	Value
dx	6.0 ft
dy	≈2 ft
dz	≈−7 ft
dps	≈0°
dth	≈−3°
alpha	≈2°
beta	≈−2°

Artemis IV booster sep flow



5. Constant-x cut through SRB noses

Color scheme:
blue: negative $C_p < 0$
red: positive C_p
pink: very high C_p
purple: $M \leq 1.0$
green: $1 < M < M_\infty$
orange: $M > M_\infty$

Variable	Value
dx	6.0 ft
dy	$\approx$2 ft
dz	≈ -7 ft
dps	$\approx 0^\circ$
dth	$\approx -3^\circ$
alpha	$\approx 2^\circ$
beta	$\approx -2^\circ$

Artemis IV booster sep flow

6. Constant-x cut through SRB nozzles

Color scheme:
blue: negative $C_p < 0$
red: positive C_p
pink: very high C_p
purple: $M \leq 1.0$
green: $1 < M < M_\infty$
orange: $M > M_\infty$

Variable	Value
dx	6.0 ft
dy	≈2 ft
dz	≈−7 ft
dps	≈0°
dth	≈−3°
alpha	≈2°
beta	≈−2°

Booster Separation Timeline

2015
Artemis I

Artemis II
2019

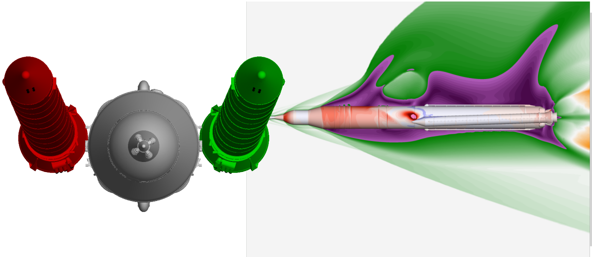
2021
Block 1B

Block 2
Beyond

- BOLE - limited supply of SRB sections
- Likely FUN3D
- Tools allow for input on design

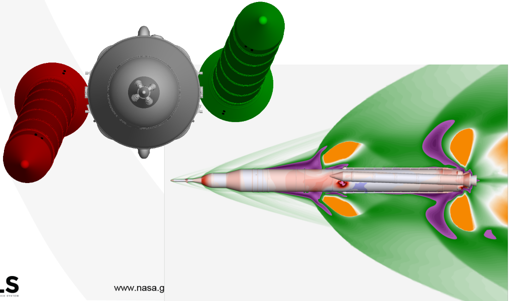
Block 2: new & improved design

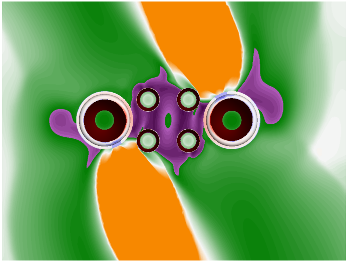
Block 1B (current design)



- For Block 2, chance to design separation that actually works for SLS
- Block 1 and 1B inherit Space Shuttle design
- Our team ran a handful of FUN3D cases to validate new design in just 2.5 weeks

Block 2 (2030 and beyond)





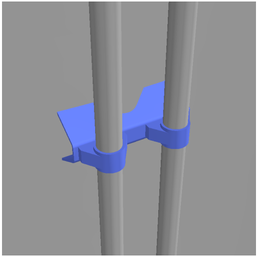
Artemis I

Preparing for Artemis I

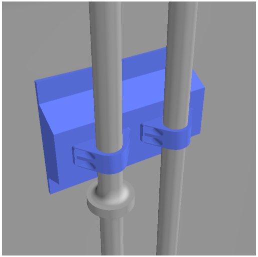
Continuously evaluating updates to the actual geometry as first flight approaches

Most of the aero work for Artemis I is at least 6 years old, so we continuously look for anything that would invalidate our older findings

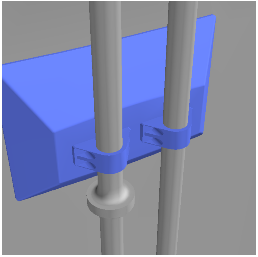
We often run Cart3D on highly detailed geometry, including bolts and things like that, as seen in the SLS-10010 geometry



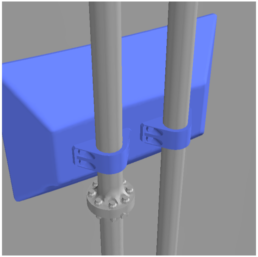
SLS-10005
2014 geometry



SLS-10008// *2016 geometry*



SLS-10009
2020 geometry

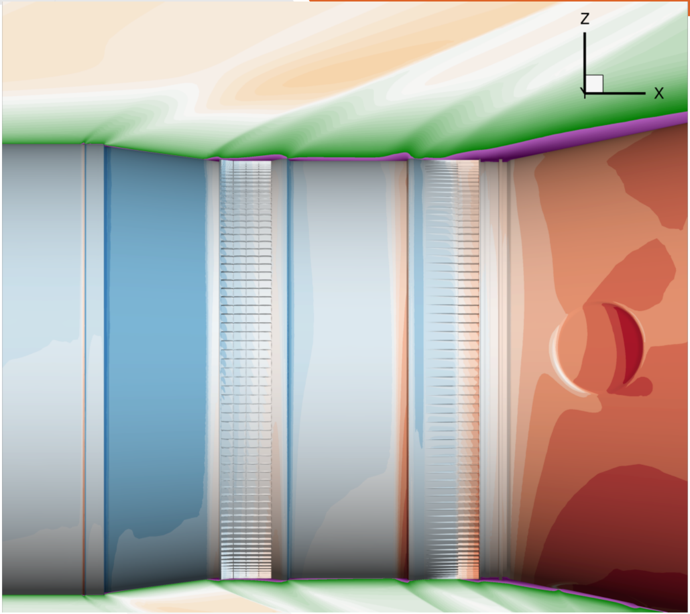


SLS-10010
2021 geometry

Preparing for Artemis I: ICPS

Artemis II ICPS
https://www.nasa.gov/exploration/systems/sls/multimedia/images.html

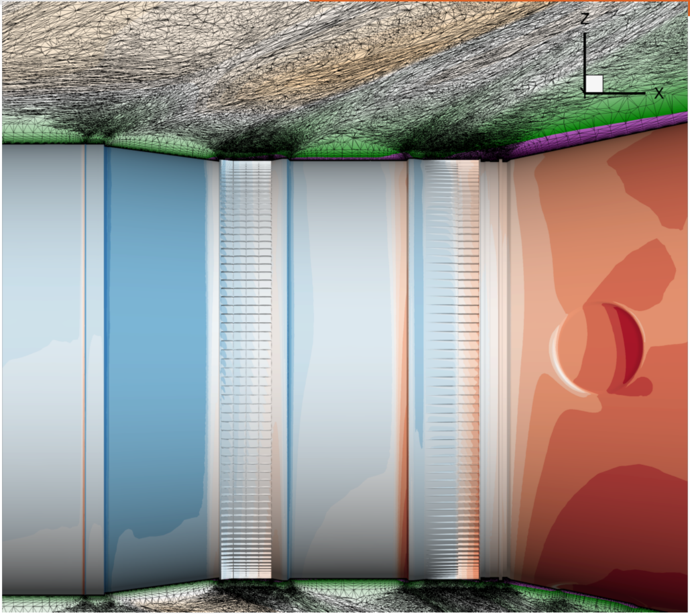
Preparing for Artemis I: ICPS



Continuing to run all our tools on *very* complex geometries

We never stop learning about the SLS flight environment, even if the data from 8 years ago remains valid for engineering purposes

Preparing for Artemis I: ICPS

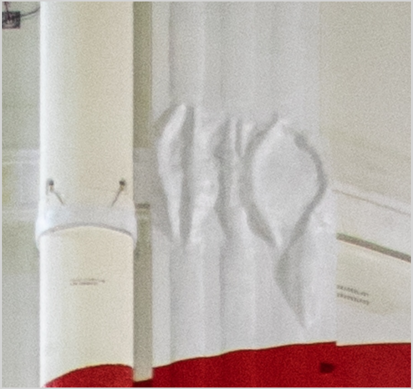


Continuing to run all our tools on *very* complex geometries

We never stop learning about the SLS flight environment, even if the data from 8 years ago remains valid for engineering purposes

The Real as-built SLS

If you look closely, some areas of the real vehicle look very . . . different from our models



Acknowledgments

- Previous members of the NASA ARC/TNA SLS CFD Team
- Other current and former members of the SLS Aero Task Team
- NASA Advanced Supercomputing facilities
- SLS Program; this work is part of the SLS Aero Task Team
- Exploration Systems Development Mission Directorate (ESDMD)
- NASA Engineering & Safety Center (NESC) for discussions and reviews